

Sept 6

Implementing GS

Initialization $\leftarrow \leq T_0$

while (...) \leftarrow # iterations = $T_1 \leq n^2$

Body $\leftarrow \leq T_2$

Output S $\leftarrow \leq T_3$

Lemma 1
from
last 2
lectures

Overall runtime $\leq T_0 + T_1 \cdot T_2 + T_3$
 $T_1 \leq n^2 \rightarrow \leq T_0 + n^2 \cdot T_2 + T_3$

IF we can
assume
 $T_0, T_3 \leq O(n^2)$
 $T_2 \leq O(1)$

$\rightarrow \leq O(n^2) + n^2 \cdot O(1) + O(n^2)$
 $\leq O(n^2) + O(n^2) + O(n^2) \leq O(n^2)$

Notation change : $M = \{1, \dots, n\} \stackrel{\text{def}}{=} [n]$

$W = [n]$

$\{w_1, \dots, w_n\} \mapsto \{1, \dots, n\}$
 $\{m_1, \dots, m_n\} \mapsto \{1, \dots, n\}$

} like our solution
Q1(a) on HW 0

\rightarrow Array indices start from 1

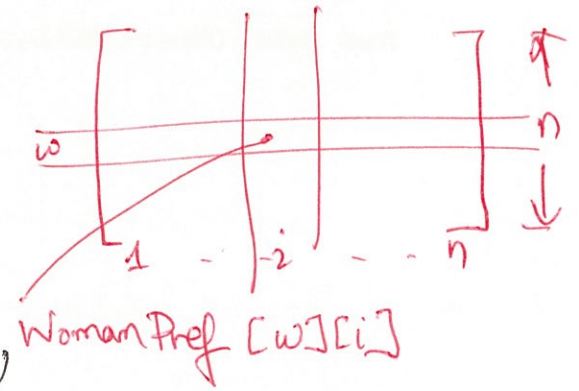
Q0) How is the input represented?

2D arrays : ~~to~~ Woman Pref, ManPref

WomanPref [W][i] = ID of the its most preferred man in L_w

ManPref [M][i] = ID

— woman in L_m



Initialization: —

Query: Read \swarrow : $O(1)$ time.

Update: —

(Q1) How do we find a free woman w ?

A1) Maintain a linked list free (of free women)

Init: Add all n women to free : $O(n)$

Query: Pick (say) the 1st element in free (+ delete it from front of free) $O(1)$

Update: w proposes to m

$O(1)$ {

Case 1: m was free \Rightarrow do nothing

Case 2.1: (m, w') remain engaged : add w to free : $O(1)$

Case 2.2: (m, w) get engaged : add w' to free

(Q2) How would w pick her best unproposed man m

A2) Maintain an array next of size n

$next[w] =$ rank of the man m that w should propose to

Q: What is the ID? Query $WomanPref[w][next[w]]$

Init: $next[w] = 1 \forall w : O(n)$

Query: Who should w propose to? $O(1)$

Update: $next[w] = next[w] + 1 \Leftarrow O(1)$

| |
|-------------------------------------------|
| So far: |
| <u>Init</u> : $O(n) + O(n) + O(n) = O(n)$ |

| |
|-----------------------------------------------|
| Query/update = $O(1) + O(1) + O(1) \leq O(1)$ |
|-----------------------------------------------|

(Q3) How do we know who m is engaged to?

A3) Array current of size n

$current[m] = \begin{cases} -1 & \text{if } m \text{ is free} \\ w & \text{if } (m, w) \text{ is engaged} \end{cases}$

Init: $current[m] = -1 \forall m : O(n)$

Query: Read $current[m] : O(1)$

Update: If after w 's proposal, (m, w) are engaged \Rightarrow $current[m] = w \rightarrow O(1)$

Q4) If $w' > w$ in L_m ?

A4') Scan $\text{ManPref}[m]$

Compute i' & i

→ check if $i' < i$?

↑ $O(1)$

$O(n)$

⇒ Query time is $O(n)$

⇒ $O(n^3)$ overall

